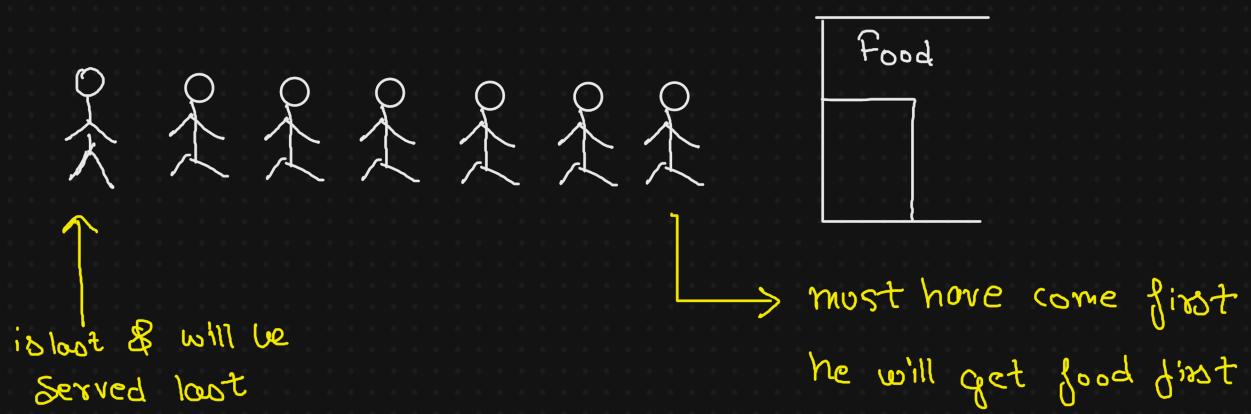


Queue

Queue is a Queue

Linear Data Structure



in → _____ → Out



LIFO

FIFO



Toll Plaza

FIFO → First in First out

Real life example

- Printer Jobs
- Customer Support
- Elevator request
- Task Scheduling

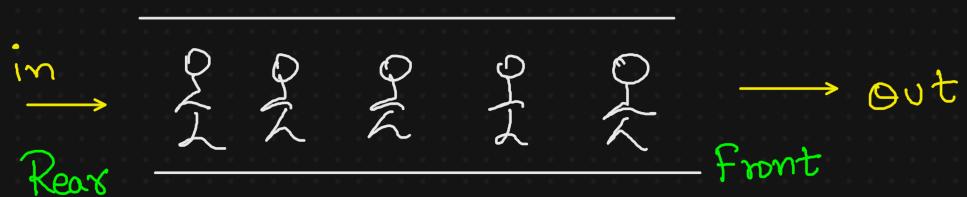
Industry example

- Job scheduling
- Network traffic
- Messaging Service → RabbitMQ, DDS, Kafka
- BFS example

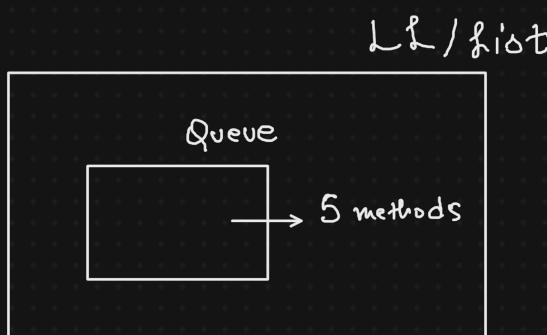
Queue Implementation and Operations

Abstract Data Type

it can be implemented using a list or a linked list



1. enqueue - to add an element into our queue
2. dequeue - when we remove an element from our queue
3. size - tell no. of elements in our queue
4. isEmpty - Queue empty or not
5. Front - return the element next to come out.



Queue Implementation : Using Inbuilt List

1. Enqueue $\ell1.append()$ $\xrightarrow{\text{in}}$ $\underline{\quad \quad}$ $\longrightarrow \text{out}$
2. Front $\ell1[0]$ rear front
3. dequeue $\ell1.pop(0)$
 ↓
 index $\ell1 = [\quad 2 \quad 3 \quad]$
 0 1
4. Size $\text{len}(\ell1)$
5. isEmpty $\text{size} == 0$

FIFO

Queue using linked list



1. enqueue: possible in O(1)

we will have to maintain a tail



2. Front(): head.next O(1)

3. dequeue(): head = head.next O(1)

4. size(): will have to maintain a len variable, for O(1) approach

5. isEmpty(): len == 0 or head is None O(1)

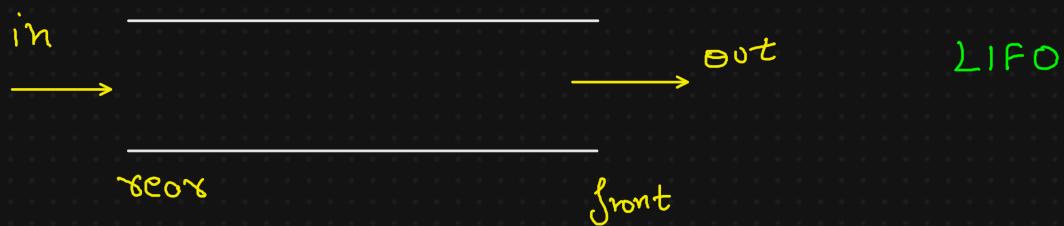


tail = None

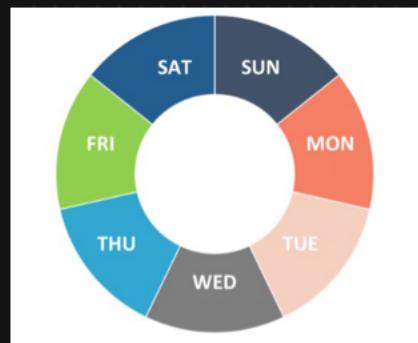
on dequeue, head = head.next
= None

Types of Queues

1. Simple Queue → single entry and single exit



2. Circular Queue → last node is connected to my first node



Ring Buffer

3. Priority Queue → elements are served / popped based on the priority



restaurant reservation

not first-come first-serve

4. Double ended Queue = DeQueue

